

You will need to obtain the signature of your instructor or TA on the following items in order to receive credit for your lab assignment. Print your name below, sign the honor code pledge, circle your course number, and then demonstrate your working hardware & firmware in order to obtain the necessary signatures.

Student Name: SHRUTHI THALLAPALLY

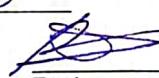
Honor Code Pledge: "On my honor, as a University of Colorado student, I have neither given nor received unauthorized assistance on this work. I have clearly acknowledged work that is not my own."

Student Signature: T. Shruthi

Signoff Checklist

Part 1 Elements

- Schematic of acceptable quality (all components shown)
- Pins and signals labeled, decoupling capacitors, and two 28-pin wire wrap sockets present on board
- Very good knowledge of a terminal emulator
- Demonstrates all 32KB of XRAM in memory map are functional, including monitor block fill command
- Using PAULMON2, demonstrates highest baud rate as: 57600
- Knows how to use SDCC [IDE or make optional]

 10/20/2023

TA signature and date

Part 2 Elements

- Knows how to analyze output files (.RST, .MEM, .MAP) for correct addresses
- C serial program and virtual debug port functional and code commented
- Hex display of buffer contents

 10/20/2023

TA signature and date

Part 3 Required and Supplemental Elements

- Required ARM code integration and execution
- 8051 PWM control works correctly, X2 mode
- Correctly enters Idle mode and exits via external interrupt 1
- Correctly enters Power Down mode
- All other PCA software menu items function correctly
- Good understanding of PCA modes
- Good user interface; program is easy to use

 27/10/23

TA signature and date

Instructor/TA Comments:

FOR INSTRUCTOR USE ONLY	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Part 1 and 2 Elements					
Schematics, SPLD code	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Hardware physical implementation	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Part 1 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sign-off done without excessive retries	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 2 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FOR INSTRUCTOR USE ONLY	Not Applicable	Below Expectation	Meets Requirements	Exceeds Requirements	Outstanding
Part 3 Elements					
Part 3 Required Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Supplemental Elements functionality	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Student understanding and skills	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall Demo Quality (Part 3 elements)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comments:

- Optional Challenge: PAULMON2 RUN command
- Optional Challenge: ISP API calls
- Optional Challenge: C and Assembly interfacing
- Optional Challenge: Serial ISR
- Optional Challenge: SDCC heap memory management analysis

Part 1 & 2

0xXXXX: 0xA1 0xB3

- (+) Correct schematic.
- (+) Correct interfacing of NVSRAM.
- (+) SDO/Codesblocks used.
- (+) Code and UI is generally functional and stable.
 - (-) '+' not implemented.
 - (-) Hex dump format not correct.
 - (-) Virtual Port not implemented.
 - (-) Needed help with .map file.
- (+) Code that is functional is of good quality.

Part 3

- (+) ARM code functional with PB & UART & mapped to PD12.
- (+) PWM functional with 33% duty cycle.
- (+) Implemented Min & Max freq.
- (+) Implemented High speed o/p.
- (-) UI can be improved.
- (-) Idle & PWR down not implemented

Submission Questions:

18.

a) What operating system (including revision) did you use for your code development?

Ans: Windows 11

b) What compiler (including revision) did you use?

Ans: SDCC

c) What exactly (include name/revision if appropriate) did you use to build your code (what IDE, make/makefile, or command line)?

Ans:STM32CubeIDE 1.13.2 and CodeBlocks:SDCC

d) Did you install and use any other software tools to complete your lab assignment?

Ans No.

e) Did you experience any problems with any of the software tools? If so, describe the problems.

Ans: No

Pictures/Screenshots:

1. Part 1: Terminal output after filling the data segment with a "U" character (55).

Data segment:

COM11 - Tera Term VT

File Edit Setup Control Window Help

Welcome to PAULMON2 v2.1, by Paul Stoffregen

See PAULMON2.DOC, PAULMON2.EQU and PAULMON2.HDR for more information.

Program Name	Location	Type
Reset	1800	External command
Single-Step	1400	External command
Memory Editor <UT100>	1800	External command

PAULMON2 Loc:2000 >
PAULMON2 Loc:2000 >
PAULMON2 Loc:2000 > Memory Editor <UT100>

DATA 8051 External Memory Editor, Paul Stoffregen, 1996
ADDR: #0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A+B+C+D+E+F ASCII EQUIVILANI

Address	Value	Character
2000: 55	55	55
2001: 55	55	55
2002: 55	55	55
2003: 55	55	55
2004: 55	55	55
2005: 55	55	55
2006: 55	55	55
2007: 55	55	55
2008: 55	55	55
2009: 55	55	55
200A: 55	55	55
200B: 55	55	55
200C: 55	55	55
200D: 55	55	55
200E: 55	55	55
200F: 55	55	55

^A=ASCII ^X=Hex ^F=Fill ^G=Goto ^C=Code ^D=Data ^L=Redraw ^Q=Quit

19°C Clear

Search

16:39 29-10-2023

Downloading a file using “D” command in PAULMON2.

Welcome to PAULMON2 v2.1, by Paul Stoffregen
See PAULMON2.DOC, PAULMON2.EQU and PAULMON2.HDR for more information.

Program Name	Location	Type
List	1000	External command
Single-Step	1400	External command
Memory Editor (UT100)	1800	External command

PAULMON2 Loc:2000 > Download
Begin ascii transfer of Intel hex file, or ESC to abort
.....
Download completed
Summary:
648 Lines received
10190 bytes received
10190 bytes written
No errors detected
PAULMON2 Loc:2000 > █

2.Part2: Creating buffers.

Creating buffers 0 and 1 with equal size with the input number which is between 32 and 4800 both inclusive.

The input number must be divisible by 16.

Welcome to PAULMON2 v2.1, by Paul Stoffregen
See PAULMON2.DOC, PAULMON2.EQU and PAULMON2.HDR for more information.

Program Name	Location	Type
List	1000	External command
Single-Step	1400	External command
Memory Editor (UT100)	1800	External command

PAULMON2 Loc:2000 > Jump to memory location
Jump to memory location <2000>, or ESC to quit: 2000
running program:
WELCOME
Enter a number between 32 and 4800 which is divisible by 16
640
entered input:640
input number:640
valid input
Memory allocated successfully for buffer 0 and buffer 1
buffer 0 starts at address: 0x3
buffer 1 starts at address: 0x285
OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.
enter the input

Terminal output when the input is a storage character.

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.
enter the input
P
OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
A
OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
L
OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
Y
OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.
```

Terminal output when the input is "+", creating new buffers whose size must be between 20 and 400.

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
Press @ to free all buffers and start program again.
enter the input
+
enter buffer size between 20 AND 400 for the new buffer
200
entered input:200
memory allocation successful for buffer 2

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
+
enter buffer size between 20 AND 400 for the new buffer
150
entered input:150
memory allocation successful for buffer 3

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

-enter the input
-enter the buffer number to be deleted
2
entered input:2
Freed buffer_2

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
```

Terminal output depicting the Heap Report when input is “?”

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
?
HEAP REPORT
Buffer 0 starts at = 0x3
Buffer 0 ends at =>x283
Buffer 0 size =640
number of storage characters in buffer 0=18
number of free spaces in buffer 0=622

Buffer 1 starts at = 0x285
Buffer 1 ends at =>x505
Buffer 1 size =640
number of storage characters in buffer 1=0
number of free spaces in buffer 1=640

Buffer 3 starts at = 0x5D1
Buffer 3 ends at =>x667
Buffer 3 size =150
number of storage characters in buffer 3=0
number of free spaces in buffer 3=150
characters in buffer 0
stored chars=18
total characters entered=23
SHRUTHITHAPELLA
heap report is done
OPTIONS
```

Terminal output depicting the Heap Report when input is “=”

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
Press @ to free all buffers and start program again.

enter the input
-
enter the buffer number to be deleted
2
entered input:2
Freed buffer_2

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
-
contents of buffer 0

storage character -- its hexadecimal representation
S--53
H--48
R--52
U--55
T--54
H--48
I--49
J--4A
I--54
H--48
A--41
L--4C
L--4C
O--4F
P--50
A--41
L--4C
L--4C
Y--59

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
```

Terminal output 2 depicting the Heap Report when input is “@”

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
number of storage characters in buffer 1=0
number of free spaces in buffer 1=640

Buffer 3 starts at = 0x5D1
Buffer 3 ends at =x667
Buffer 3 size =150
number of storage characters in buffer 3=0
number of free spaces in buffer 3=150
characters in buffer 0
stored chars=18
total characters entered=23
SHRUTHIHALAPALLY
heap report is done

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

enter the input
@buffer 0 is freed
buffer 1 is freed
buffer 2 is freed
buffer 2 is freed
all buffers are freed. Please start from the beginning

WELCOME
Enter a number between 32 and 4800 which is divisible by 16
entered input:@
input number:@
InValid input. Try again


```

Terminal output 2 when invalid inputs are given.

```
COM11 - Tera Term VT
File Edit Setup Control Window Help
entered input:@
input number:@
InValid input. Try again
20
entered input:20
input number:20
InValid input. Try again
2400
entered input:2400
input number:2400
valid input
buffer 1 malloc failed. free all malloc

WELCOME
Enter a number between 32 and 4800 which is divisible by 16
640
entered input:640
input number:640
valid input
Memory allocated successfully for buffer 0 and buffer 1
buffer 0 starts at address: 0x3
buffer 1 starts at address: 0x285

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

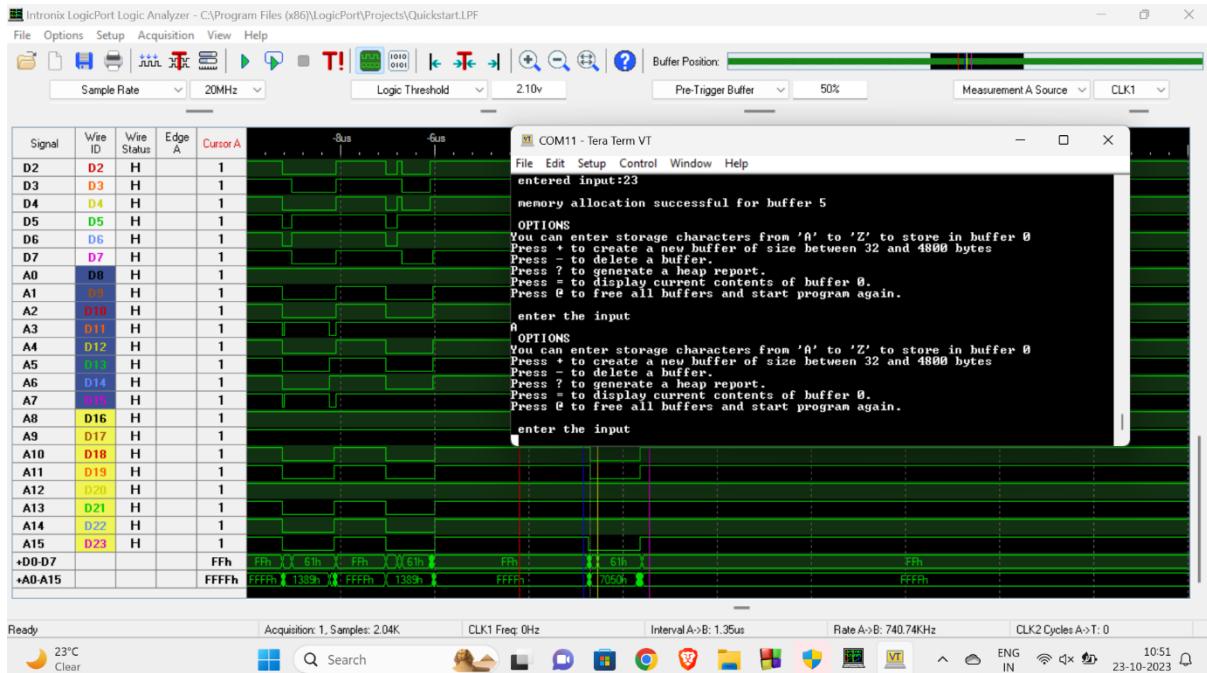
enter the input
+
enter buffer size between 20 AND 400 for the new buffer
500
entered input:500
Invalid buffer size.To try again press '+'
memory allocation failed for buffer 2

OPTIONS
You can enter storage characters from 'A' to 'Z' to store in buffer 0
Press + to create a new buffer of size between 20 and 400 bytes
Press - to delete a buffer.
Press ? to generate a heap report.
Press = to display current contents of buffer 0.
Press @ to free all buffers and start program again.

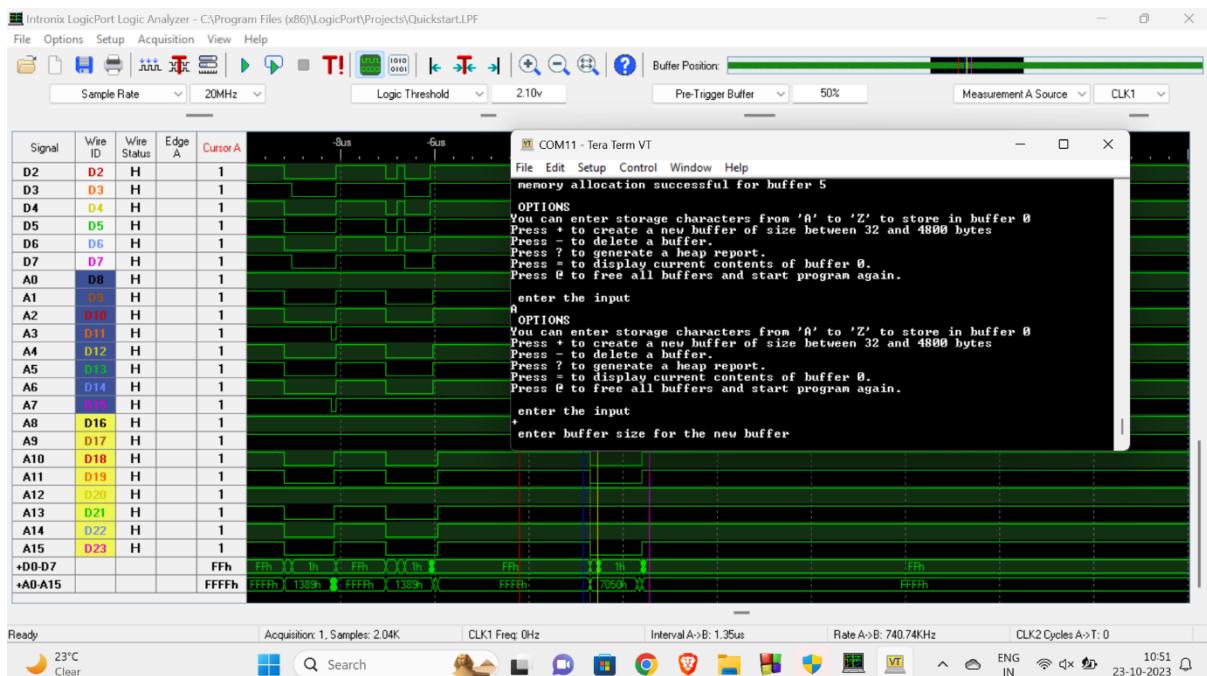
enter the input
```

3.Logic analyzer pictures of Virtual Debug port.

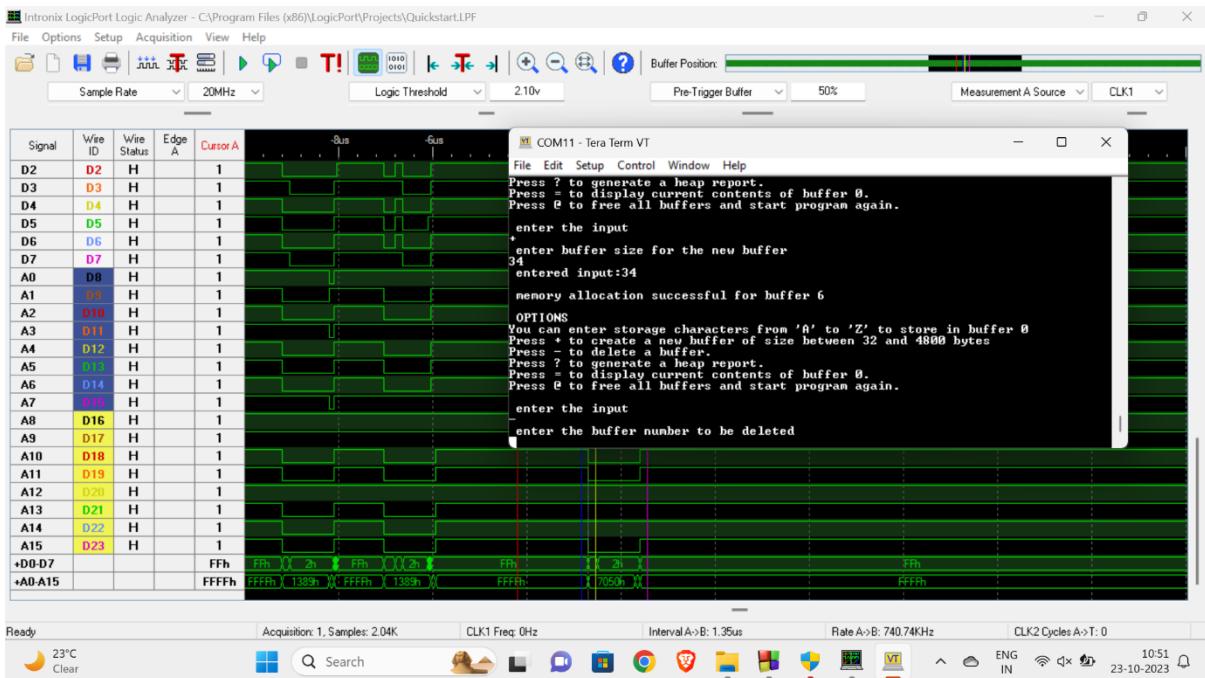
For a storage Character in Buffer 0:



For a "+" input:



For a “-“ input :



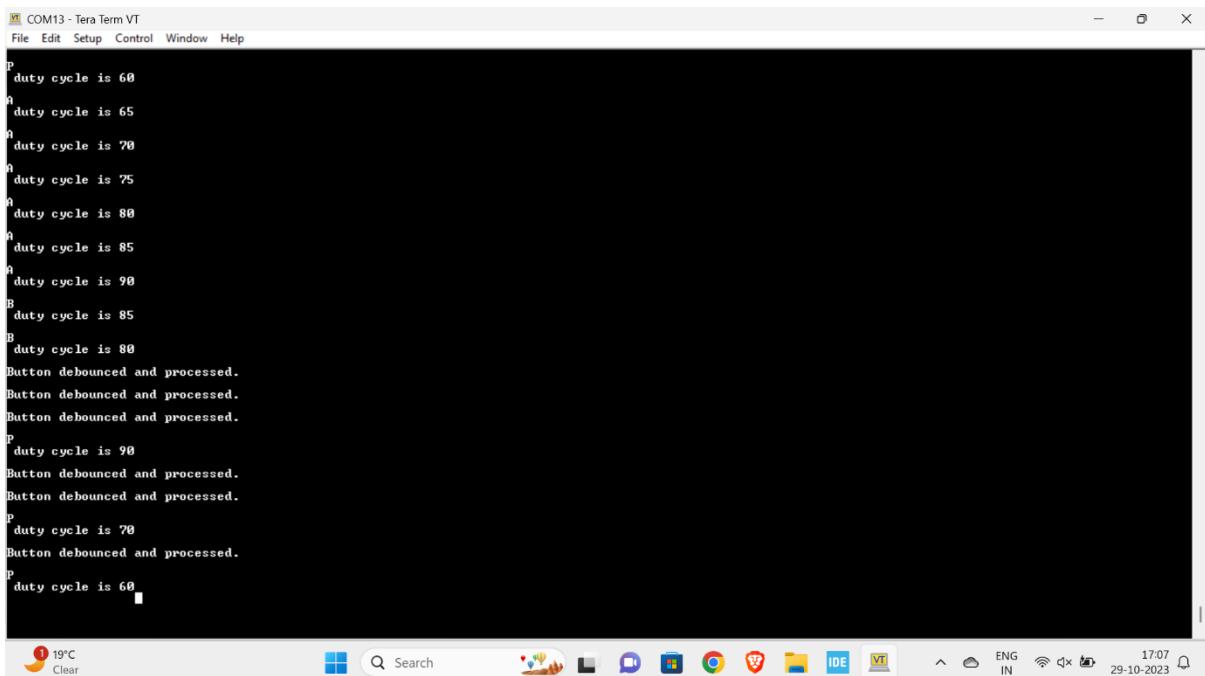
4. Part3: Terminal Pictures

For the required part, one GPIO pin is used to generate the PWM signal with a default 60% duty cycle. The duty cycle increases by 5% every time A is given as input.

The duty cycle decreases by 5% every time A is given as input.

The duty cycle increases/decreases by 10% every time the button is pressed.

When P is given as input, the current duty cycle is displayed.



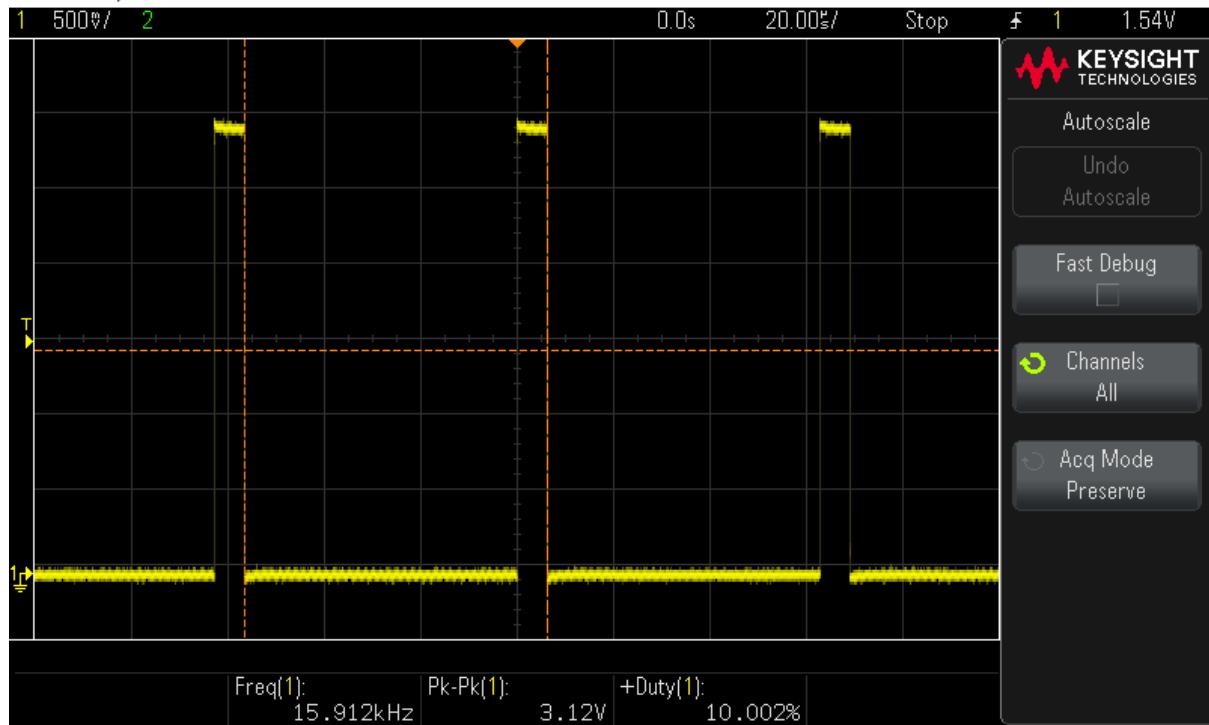
```
COM3 - Tera Term VT
File Edit Setup Control Window Help
duty cycle is 85
B
duty cycle is 80
Button debounced and processed.
Button debounced and processed.
Button debounced and processed.
P
duty cycle is 90
Button debounced and processed.
Button debounced and processed.
P
duty cycle is 70
Button debounced and processed.
P
duty cycle is 60
Button debounced and processed.
P
duty cycle is 0
B
duty cycle is 0
B
duty cycle is 0
Button debounced and processed.
P
duty cycle is 10
```

19°C Clear 17:08 29-10-2023

Oscilloscope pictures for different duty cycles varied by giving "A","B" and button.

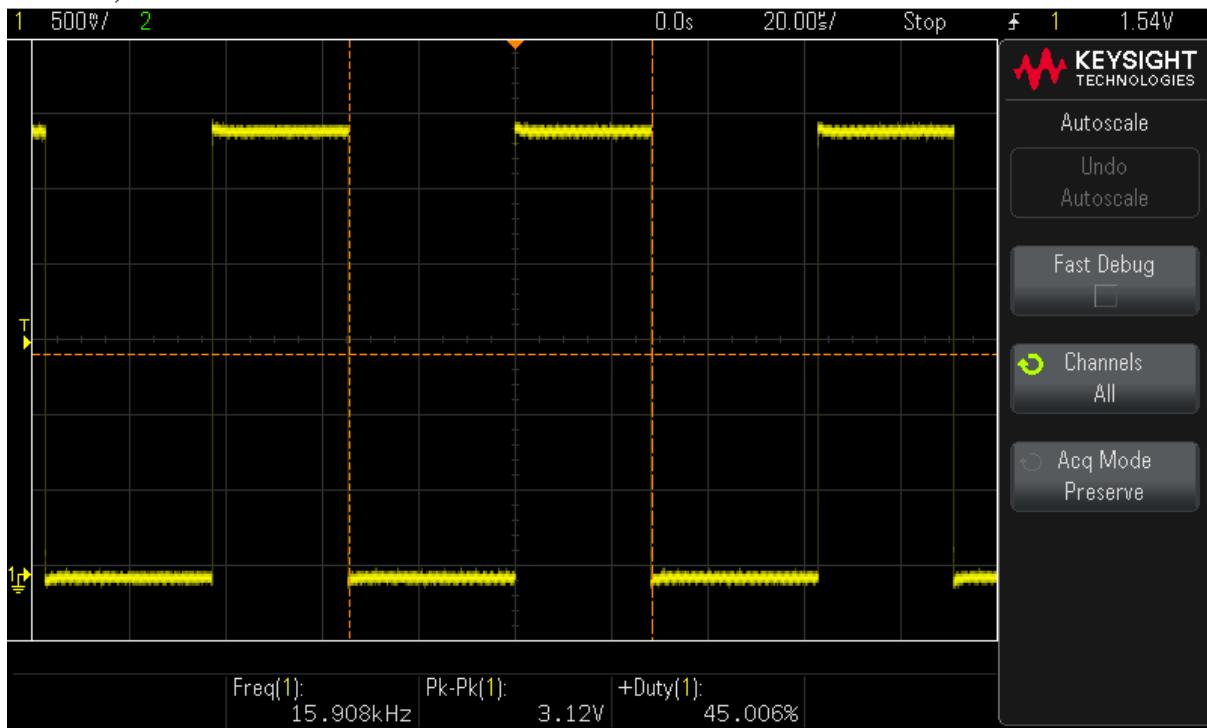
Duty cycle=10%

DSO-X 1102G, CN57266514: Mon Oct 30 06:52:29 2023



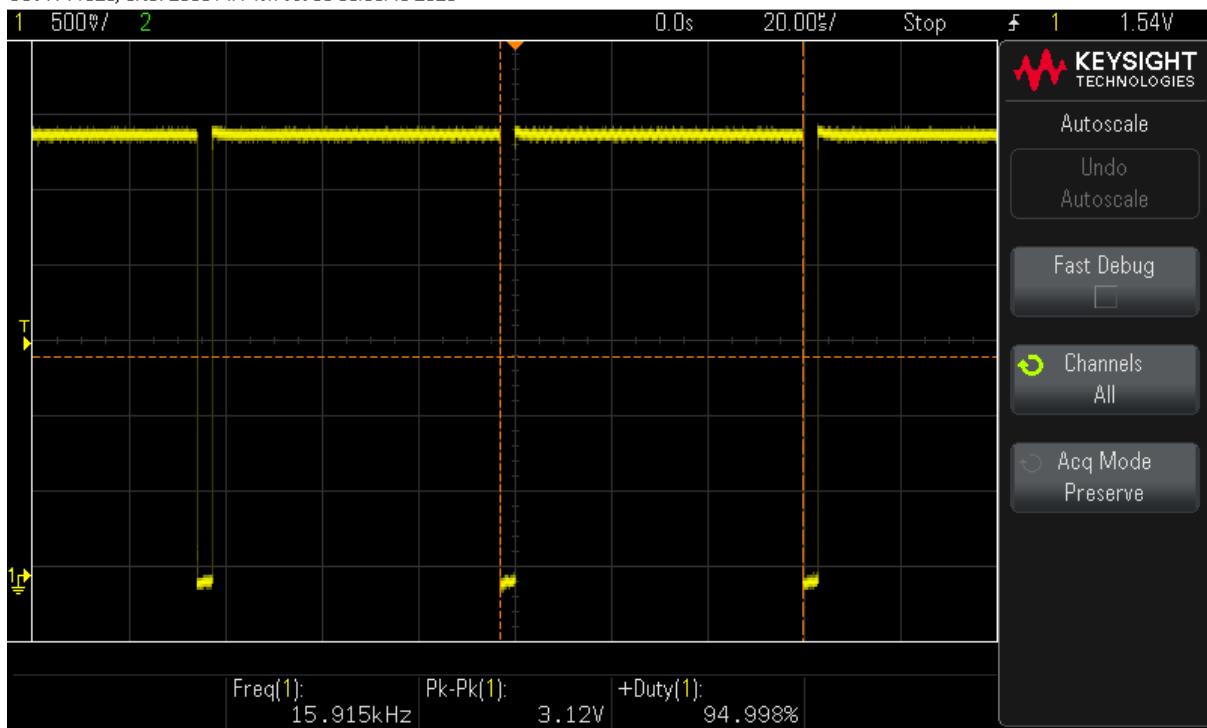
Duty cycle= 45%

DSO-X 1102G, CN57266514: Mon Oct 30 06:53:08 2023



Duty cycle=95%

DSO-X 1102G, CN57266514: Mon Oct 30 06:53:45 2023

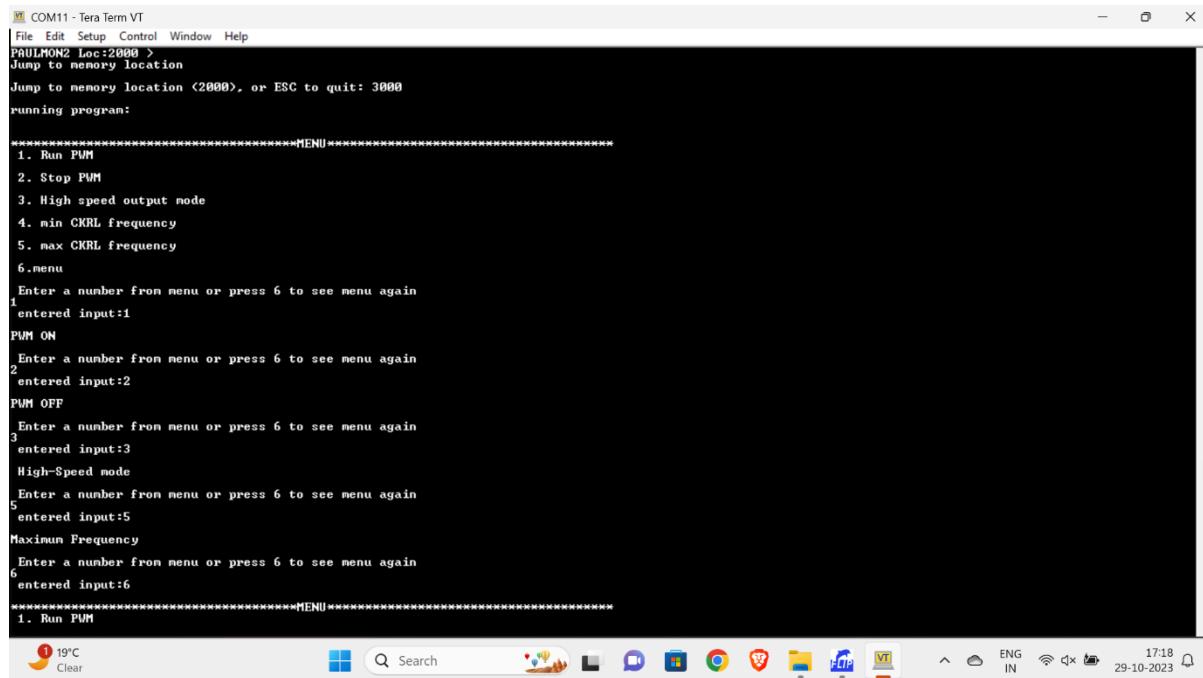


5. Terminal Pictures:

For supplemental elements, demonstrating the PCA modes.

-Pulse Width Modulation

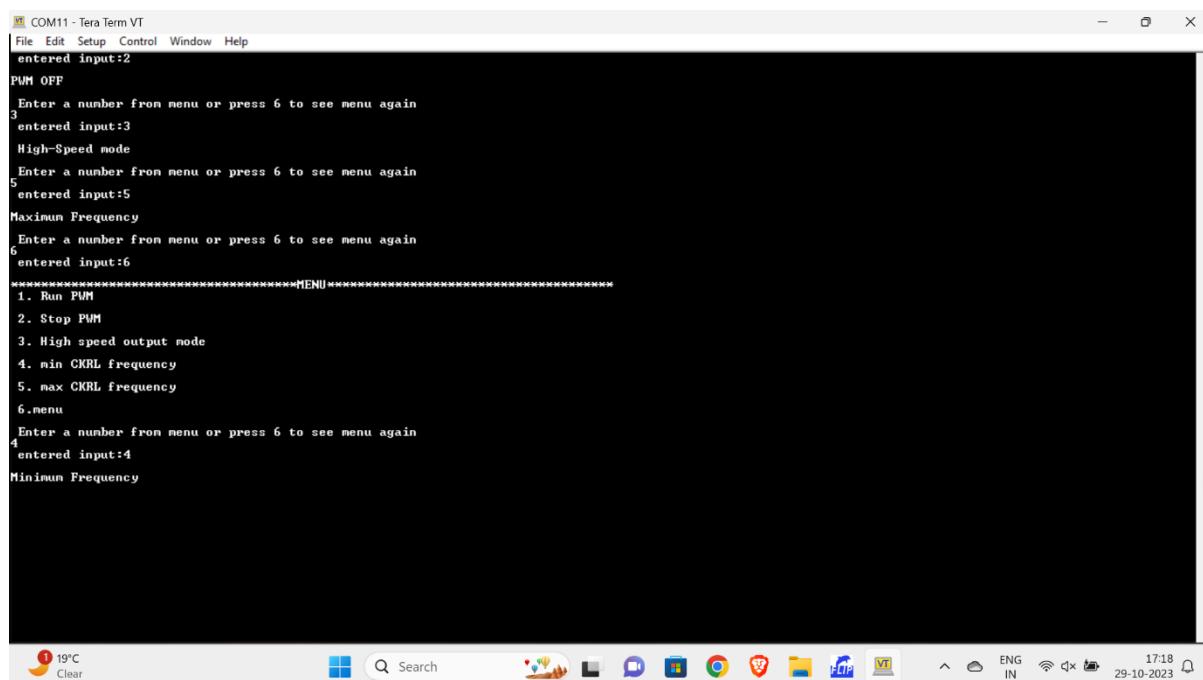
-High-Speed output



```
COM11 - Tera Term VT
File Edit Setup Control Window Help
PAULMON2 Loc:2000 >
Jump to memory location
Jump to memory location (2000), or ESC to quit: 3000
running program:

*****MENU*****
1. Run PWM
2. Stop PWM
3. High speed output mode
4. min CKRL frequency
5. max CKRL frequency
6.menu
Enter a number from menu or press 6 to see menu again
1 entered input:1
PUM ON
Enter a number from menu or press 6 to see menu again
2 entered input:2
PUM OFF
Enter a number from menu or press 6 to see menu again
3 entered input:3
High-Speed mode
Enter a number from menu or press 6 to see menu again
5 entered input:5
Maximum Frequency
Enter a number from menu or press 6 to see menu again
6 entered input:6
*****MENU*****
1. Run PWM
2. Stop PWM
3. High speed output mode
4. min CKRL frequency
5. max CKRL frequency
6.menu
Enter a number from menu or press 6 to see menu again
4 entered input:4
Minimum Frequency

19°C Clear Search ENG IN 17:18 29-10-2023
```



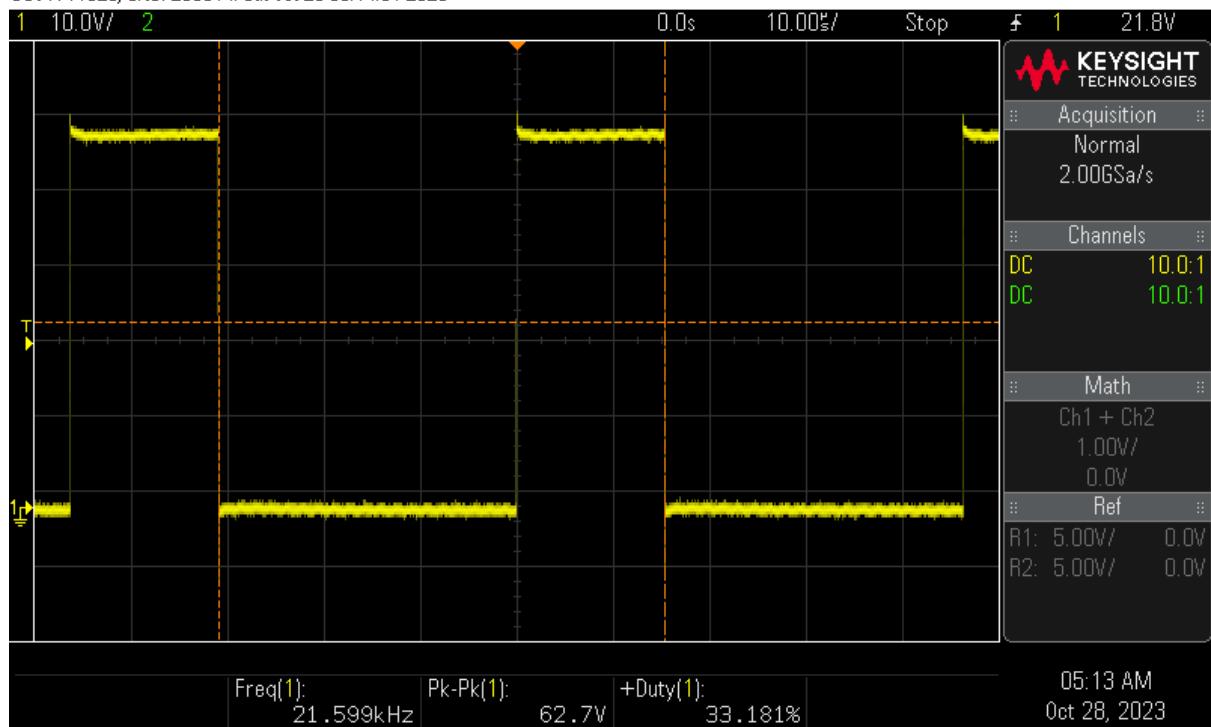
```
COM11 - Tera Term VT
File Edit Setup Control Window Help
entered input:2
PWM OFF
Enter a number from menu or press 6 to see menu again
3 entered input:3
High-Speed mode
Enter a number from menu or press 6 to see menu again
5 entered input:5
Maximum Frequency
Enter a number from menu or press 6 to see menu again
6 entered input:6
*****MENU*****
1. Run PWM
2. Stop PWM
3. High speed output mode
4. min CKRL frequency
5. max CKRL frequency
6.menu
Enter a number from menu or press 6 to see menu again
4 entered input:4
Minimum Frequency

19°C Clear Search ENG IN 17:18 29-10-2023
```

Oscilloscope pictures:

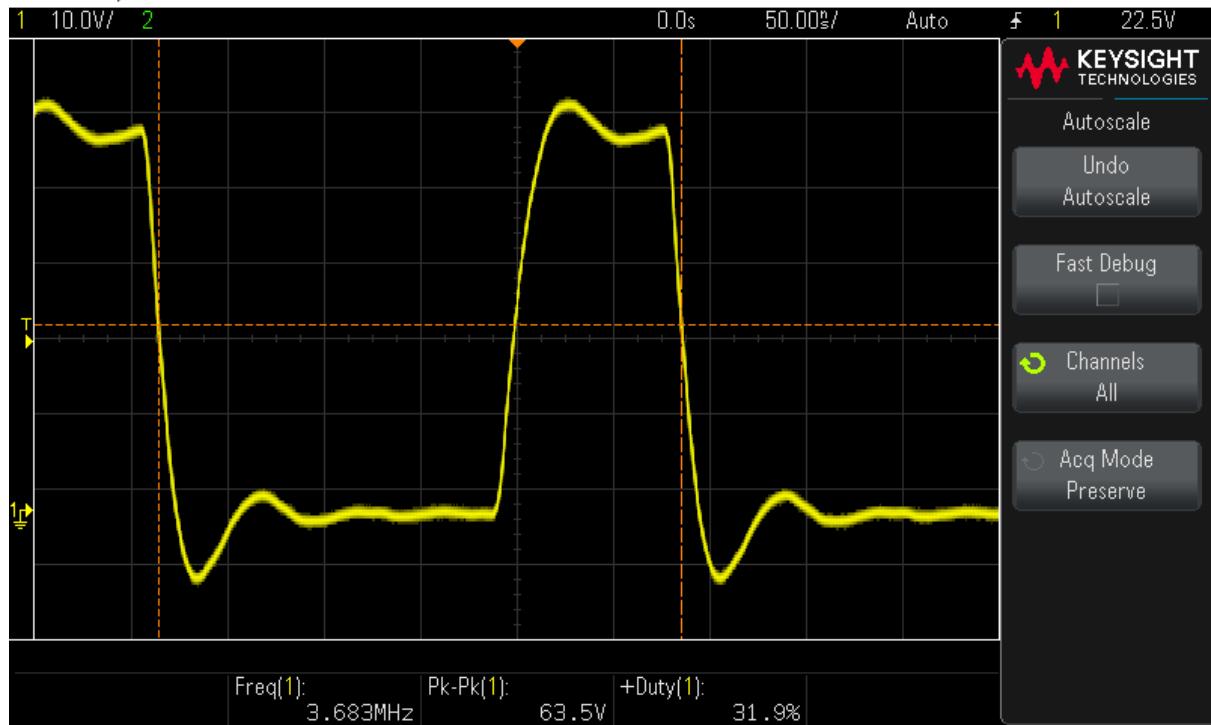
For the supplementary part, one GPIO pin is configured for PWM duty cycle of 33%.

DSO-X 1102G, CN57266514: Sat Oct 28 05:14:01 2023

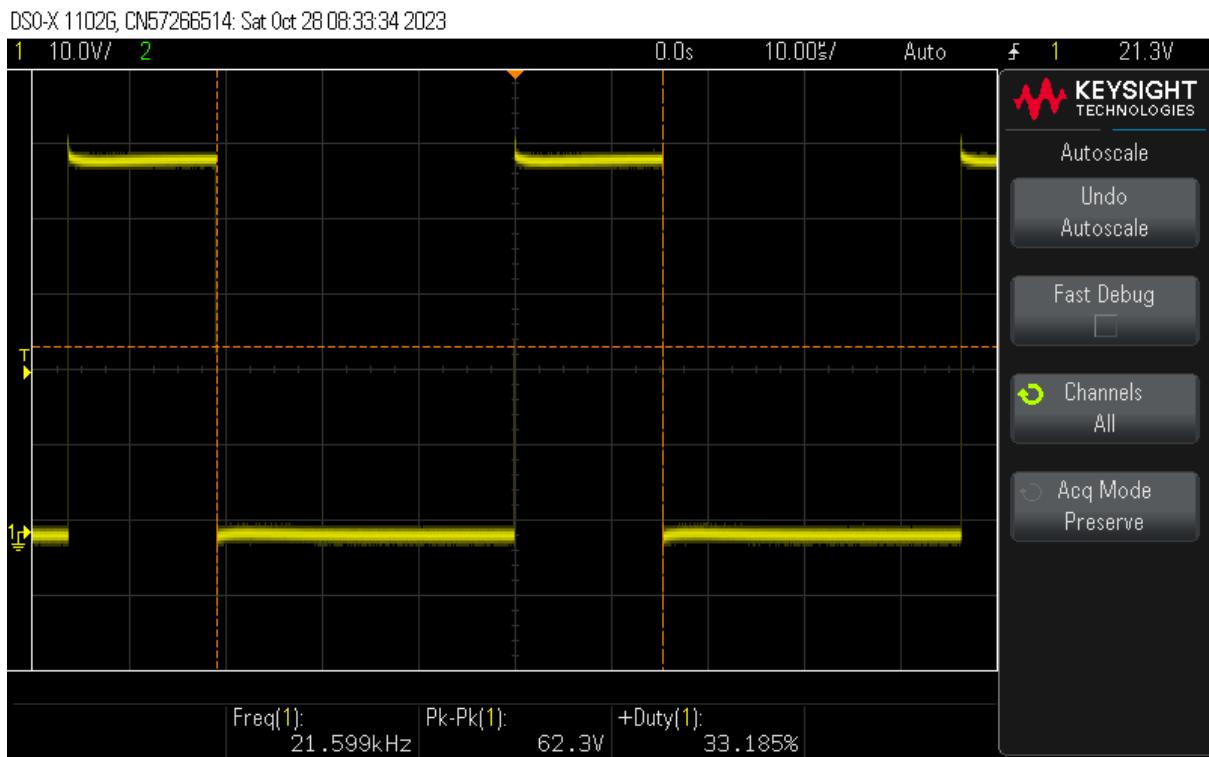


For the supplementary part, one pin is set for High-Speed Output.

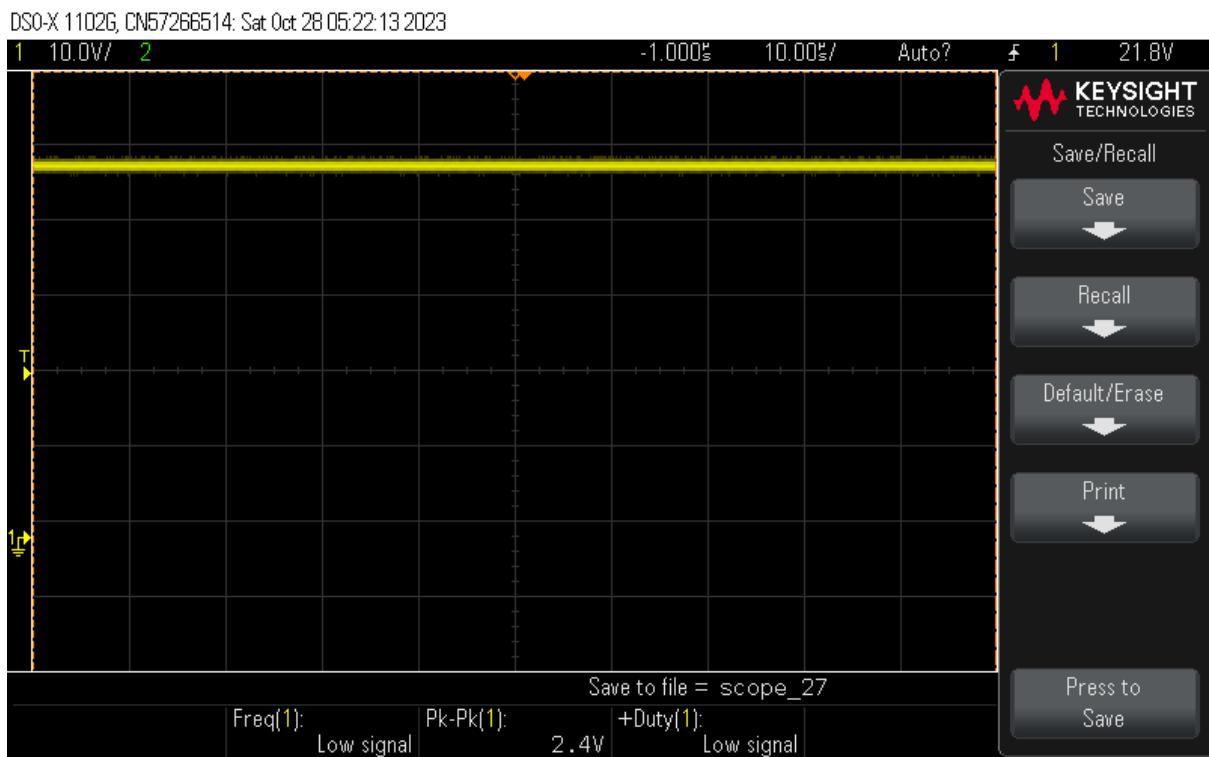
DSO-X 1102G, CN57266514: Sat Oct 28 08:34:08 2023



For the supplementary part, when PWM is ON for one pin.

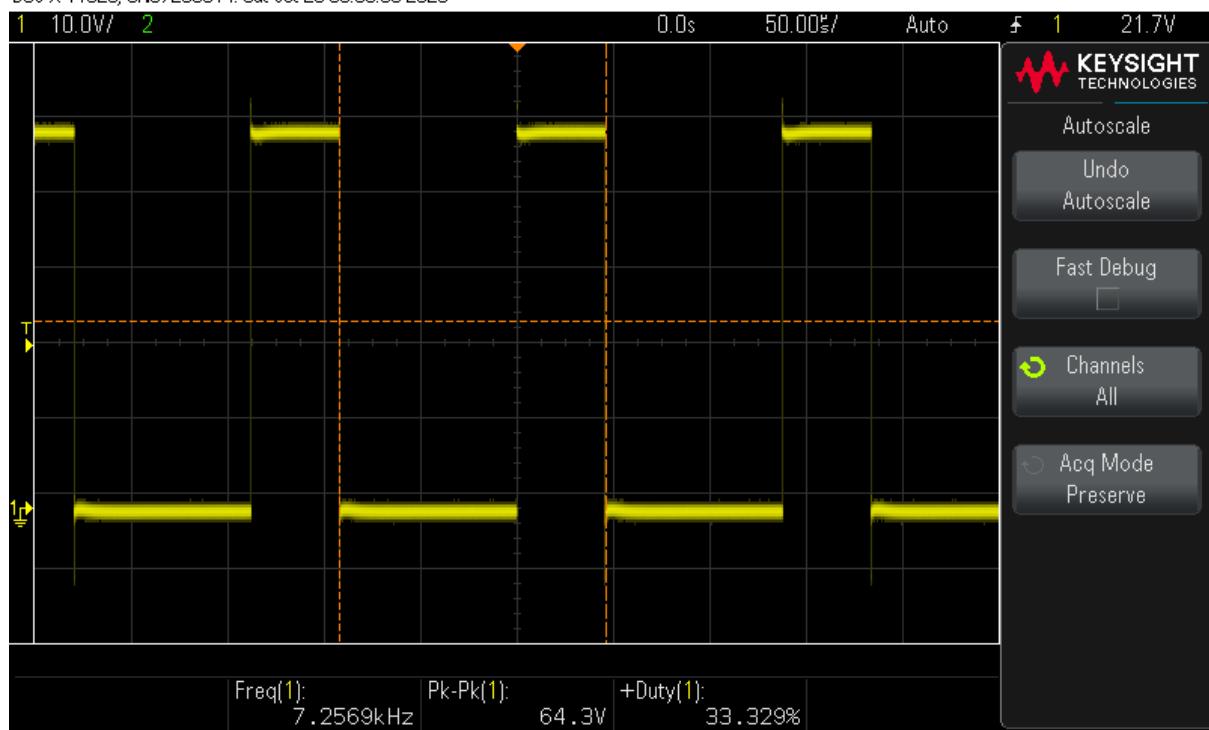


For the supplementary part, when PWM is OFF for one pin.



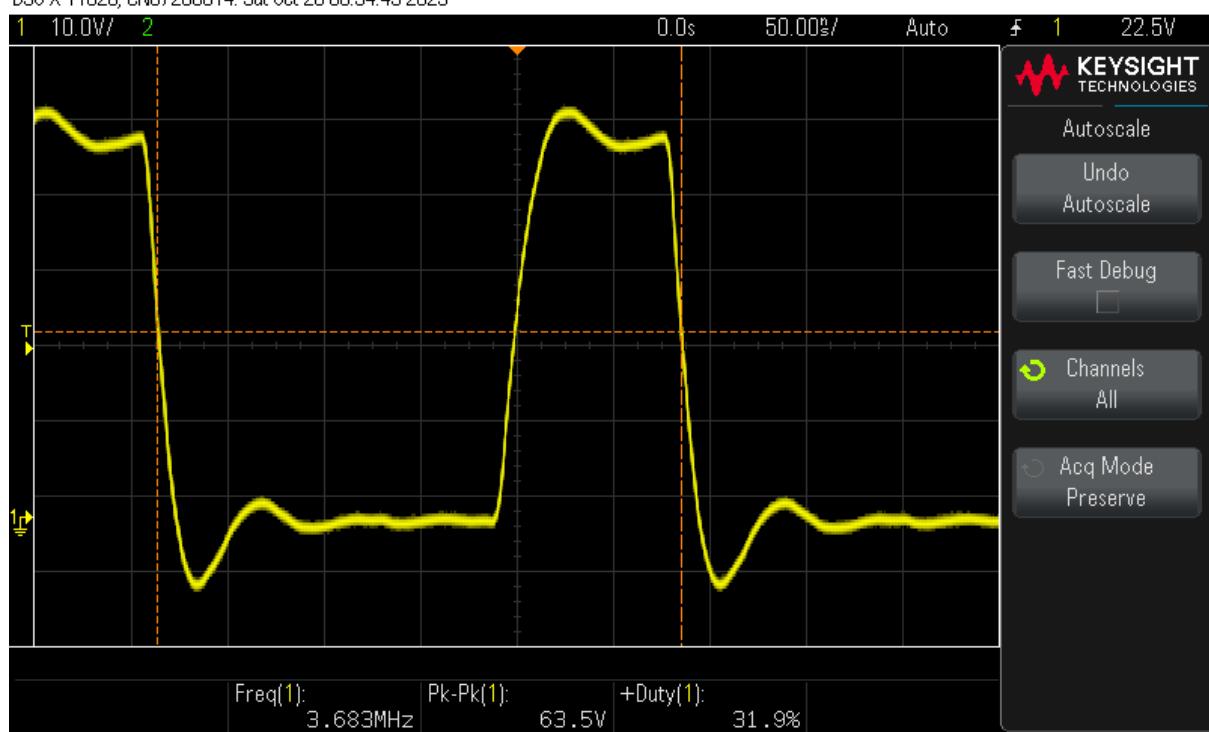
the supplementary part, output at ALE when one pin is set to the minimum peripheral clock frequency supported by the CKRL register.

DSO-X 1102G, CN57266514: Sat Oct 28 08:35:55 2023



For the supplementary part, output at ALE when one pin is set to the maximum peripheral clock frequency supported by the CKRL register.

DSO-X 1102G, CN57266514: Sat Oct 28 08:34:43 2023



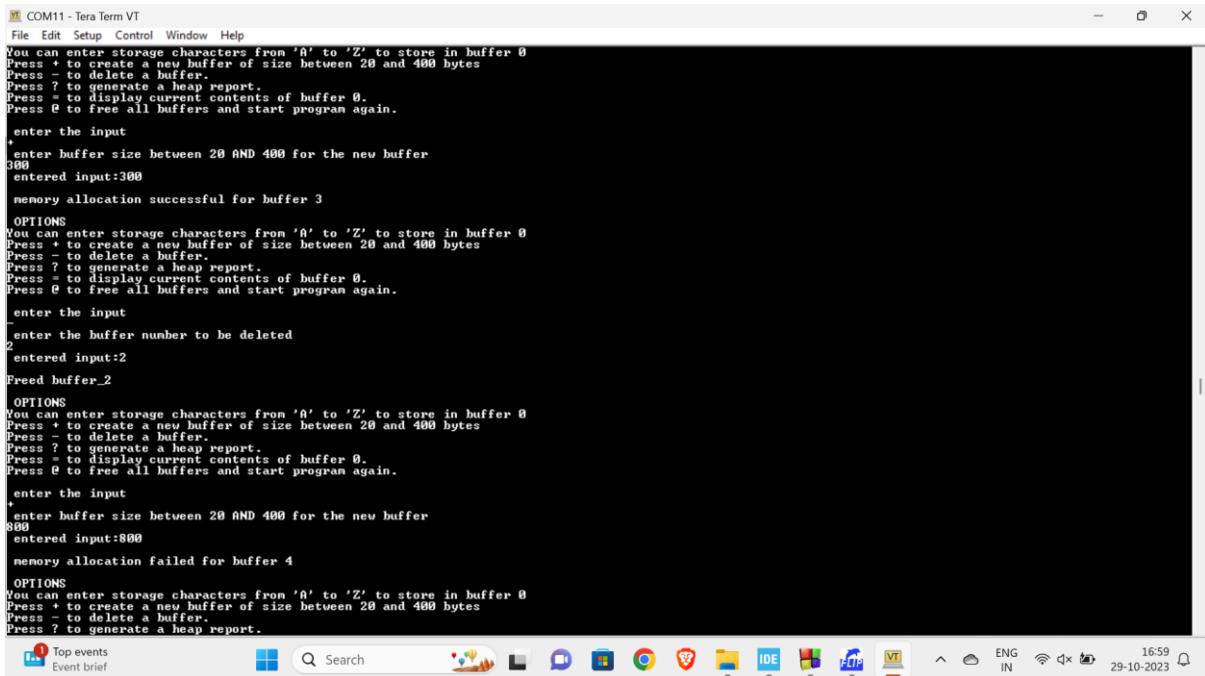
For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 100 bytes.

```
File Edit Setup Control Window Help  
HEAP REPORT  
Buffer 0 starts at = 0x963  
Buffer 0 ends at =0x963  
Buffer 0 size =2400  
number of storage characters in buffer 0=0  
number of free spaces in buffer 0=2400  
  
Buffer 1 starts at = 0x965  
Buffer 1 ends at =0x12C5  
Buffer 1 size =2400  
number of storage characters in buffer 1=0  
number of free spaces in buffer 1=2400  
  
Buffer 2 starts at = 0x1391  
Buffer 2 ends at =0x14BD  
Buffer 2 size =300  
number of storage characters in buffer 2=0  
number of free spaces in buffer 2=300  
  
Buffer 3 starts at = 0x1391  
Buffer 3 ends at =0x14BD  
Buffer 3 size =100  
number of storage characters in buffer 3=0  
number of free spaces in buffer 3=100  
characters in buffer 0  
stored chars=0
```

For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 210 bytes.

```
File Edit Setup Control Window Help  
Buffer 1 starts at = 0x965  
Buffer 1 ends at =0x965  
Buffer 1 size =2400  
number of storage characters in buffer 1=0  
number of free spaces in buffer 1=2400  
  
Buffer 2 starts at = 0x1391  
Buffer 2 ends at =0x14BD  
Buffer 2 size =300  
number of storage characters in buffer 2=0  
number of free spaces in buffer 2=300  
  
Buffer 3 starts at = 0x1391  
Buffer 3 ends at =0x14BD  
Buffer 3 size =210  
number of storage characters in buffer 3=0  
number of free spaces in buffer 3=210  
characters in buffer 0  
stored chars=0  
total characters entered=3  
heap report is done  
  
OPTIONS  
You can enter storage characters from 'A' to 'Z' to store in buffer 0  
Press + to create a new buffer of size between 20 and 400 bytes  
Press < to move to previous buffer.  
Press ? to generate a heap report.  
Press = to display current contents of buffer 0.  
Press @ to free all buffers and start program again.  
enter the input
```

For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 800 bytes.



The screenshot shows a terminal window titled "COM11 - Tera Term VT". The window displays a series of commands and their outputs related to memory allocation and deallocation. It includes help text for the program, a command to enter input, and specific commands to create a buffer of size 300, delete buffer 2, and create a buffer of size 800. The output shows the memory allocation was successful for buffer 3 and failed for buffer 4. The terminal window is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

Code snippet which I used to avoid button debouncing.

```
/**  
 * @brief Handle the EXTI0 (External Interrupt 0) interrupt  
 * @note This function is an interrupt handler.  
 */  
void EXTI0_IRQHandler(void) {  
    if (EXTI->PR & EXTI_PR1_PR0) {  
        EXTI->PR |= EXTI_PR1_PR0; // Clear pending bit for EXTI line 0 (PortA Pin 0)  
  
        if (!buttonPressed) {  
            TIM2->CR1 |= TIM_CR1_CEN; // Start Timer 2 for debouncing  
            buttonPressed = 1; // Set button press state  
        }  
    }  
}  
// + +
```

SIGNIFICANT LEARNINGS:

- I have learned how to use Paulmon2 commands to modify the data segment and navigate through the program segment.
- I have learned how to run a specific program by jumping to its location using Paulmon2 commands.
- I have learned how to write a User Interface for dynamic memory allocation of buffers in heap memory and perform various functions on it.
- I have learned how to implement PWM using a GPIO pin and change the LED intensity using PWM.
- I have learned about different PCA modes while implementing the supplementary and challenge elements.

